EXHIBIT 10

UNITED STATES PATENT NO. 11,805,267

CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

As demonstrated in the chart below, Hisense directly and indirectly infringes at least claim 19 of US 11,805,267 (the "'267 Patent"). Hisense directly infringes, contributes to the infringement of, and/or induces infringement of the '267 Patent by making, using, selling, offering for sale, and/or importing into the United States the Accused Products that are covered by one or more claims of the '267 Patent. The Accused Products are devices that decode H.265-compliant video. For example, The Hisense 43A7N is a representative product for other Hisense devices that decode H.265-compliant video.

The Hisense 43A7N contains at least one video decoder that helps decode H.265-compliant video.[1] While evidence from the Hisense 43A7N is specifically charted herein, the evidence and contentions charted herein apply equally to the other Hisense Accused Products that decode H.265-compliant video.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the '267 Patent. Moreover, this exemplary chart does not limit, and is not intended to limit, Nokia's infringement positions or contentions.

The following infringement chart includes exemplary citations to ITU-T Rec. H.265 (12/2016) High efficiency video coding (available at https://www.itu.int/rec/T-REC-H.265-201612-S/en) (the "H.265 Standard"). The cited functionality has been included in editions of the H.265 Standard since April 2013 and remains in current editions of the H.265 Standard. Any Hisense device that includes a decoder that practices the functionality in any of these editions of the H.265 Standard practices ("H.265 Decoder") the claims of the '267 Patent. Thus, the Accused Products each practice the H.265 Standard and are covered by claims of the '267 Patent.

Nokia contends each of the following limitations is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents. [2]

---

[1] See, e.g., Hisense 43A7N User Manual available at https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n.

[2] This claim chart is based on the information currently available to Nokia and is intended to be exemplary in nature. Nokia reserves all rights to update and elaborate its infringement positions, including as Nokia obtains additional information during discovery.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

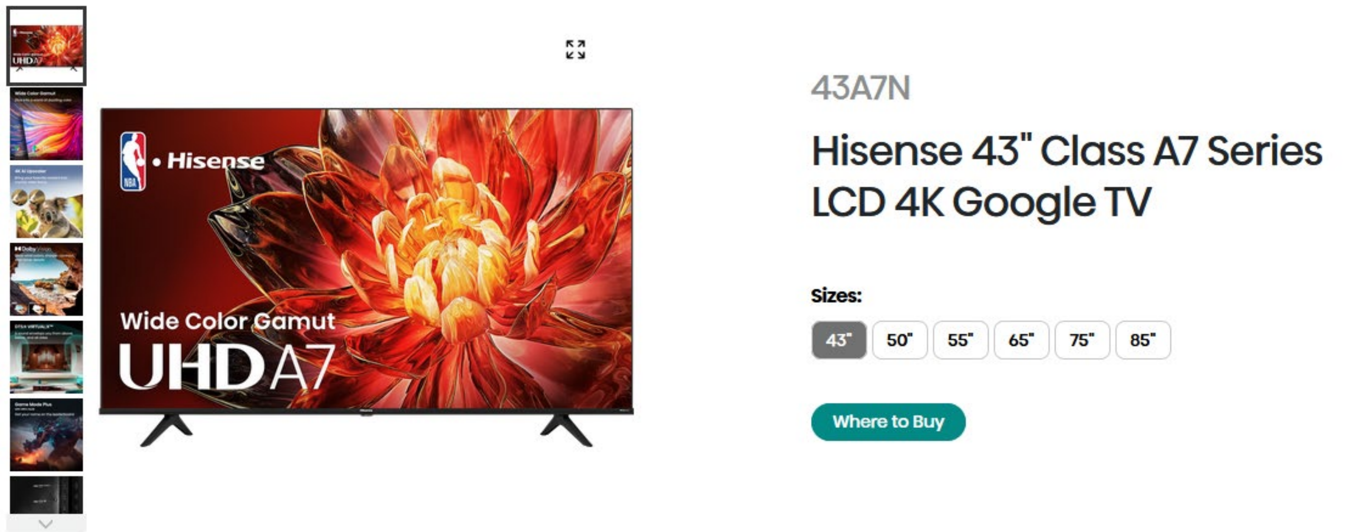
U.S. PATENT NO. 11,805,267	**HISENSE ACCUSED PRODUCTS**
19. [A] A method for decoding a block of pixels, the method comprising:	Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding a block of pixels. For example, and without limitation, the Hisense 43A7N uses hardware-accelerated video decoding and includes Mediatek MT9602 Processor.  https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	 **Display**: 42.5 in, VA, Direct LED, 3840 x 2160 pixels **Viewing angles (H/V)**: 178 ° / 178 ° **Brightness**: 275 cd/m² **Static contrast**: 4000 : 1 **Refresh rate**: 50 Hz / 60 Hz **Frame interpolation**: 120 MR (Motion Rate) **TV tuner**: Analog (NTSC/PAL/SECAM), ATSC, Clear QAM **SoC**: MediaTek MT9602 **CPU**: ARM Cortex-A53, 1500 MHz, **Cores**: 4 **Dimensions**: 963 x 560 x 74 mm **Weight**: 6.8 kg Add to compare Suggest an edit https://www.displayspecifications.com/en/model/234a3f3b

EXHIBIT 10

UNITED STATES PATENT NO. 11,805,267

CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	Video Format

Container	Video Codec	File Extension Name	Resolution and Frame Rate
MPEG program stream	MPEG1/2	.DAT, .VOB, .MPG, .MPEG	1920 x 1080 @ 60fps
	MPEG4		
	H.264		3840 x 2160 @ 60fps
MPEG transport stream	HEVC/H.265	.ts, .trp, .tp	3840 x 2160 @ 60fps
	MPEG4		1920 x 1080 @ 60fps
	H.264		3840 x 2160 @ 60fps
	VC1		1920 x 1080 @ 60fps

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	<table><tr><th>Container</th><th>Video Codec</th><th>File Extension Name</th><th>Resolution and Frame Rate</th></tr><tr><td></td><td>MPEG1/2</td><td></td><td>1920 x 1080 @ 60fps</td></tr><tr><td></td><td>AVS</td><td></td><td></td></tr><tr><td></td><td>AVS+</td><td></td><td></td></tr><tr><td></td><td>AVS2</td><td></td><td>3840 x 2160 @ 60fps</td></tr><tr><td>MP4</td><td>VP8</td><td>.mp4, .mov</td><td>1920 x 1080 @ 60fps</td></tr><tr><td></td><td>AV1</td><td></td><td>3840 x 2160 @ 60fps</td></tr><tr><td></td><td>HEVC/H.265</td><td></td><td></td></tr><tr><td></td><td>MPEG1/2</td><td></td><td>1920 x 1080 @ 60fps</td></tr><tr><td></td><td>MPEG4</td><td></td><td></td></tr><tr><td></td><td>H.263</td><td></td><td></td></tr><tr><td></td><td>H.264</td><td></td><td>3840 x 2160 @ 60fps</td></tr><tr><td></td><td>WMV3</td><td></td><td>1920 x 1080 @ 60fps</td></tr><tr><td></td><td>VC1</td><td></td><td></td></tr><tr><td></td><td>Motion JPEG</td><td></td><td>1920 x 1080 @ 30fps</td></tr></table> Source: Hisense 43A7N User Manual, at 45-46. Downloaded from https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025). For example, a Hisense 43A7N was used to play back an H.265-compliant video.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

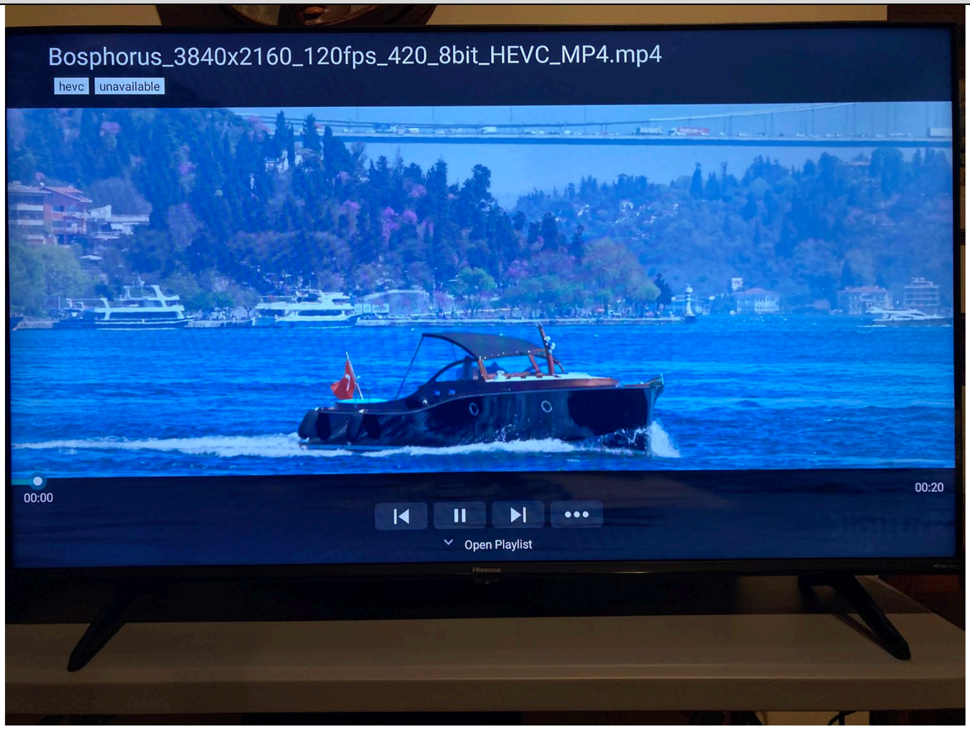
U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	 Source: Picture of H.265-complaint video playback on Hisense 43A7N For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising the limitations below. **3 Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply:

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	3.12 **bitstream**: A sequence of bits, in the form of a *NAL unit stream* or a *byte stream*, that forms the representation of *coded pictures* and associated data forming one or more coded video sequences *(CVSs)*. **3.41** **decoder**: An embodiment of a *decoding process*. **3.44** **decoding process**: The process specified in this Specification that reads a *bitstream* and derives *decoded pictures* from it. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 4-7.
[B] determining, for a current block, a first reference block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;	Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding video comprising determining, for a current block, a first reference block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision. Each of the Accused Products performs a method for decoding video comprising determining, for a current block, a first reference block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision, corresponding to the decoding process specified by the H.265 Standard. For example, as shown below a "bi-predictive (B) slice" is decoded using intra or inter prediction with at most two motion vectors and reference indicies to predict the sample values of each block. The following specifications provide further evidence of how each of the Accused Products operates: **3** **Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply: **3.11** **bi-predictive (B) slice**: A *slice* that is decoded using *intra prediction* or using *inter prediction* with at most two *motion vectors* and *reference indices* to *predict* the sample values of each *block*.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	3.12 **bitstream**: A sequence of bits, in the form of a *NAL unit stream* or a *byte stream*, that forms the representation of *coded pictures* and associated data forming one or more coded video sequences *(CVSs)*. **3.41** **decoder**: An embodiment of a *decoding process*. **3.44** **decoding process**: The process specified in this Specification that reads a *bitstream* and derives *decoded pictures* from it. **3.63** **inter coding**: Coding of a *coding block*, *slice*, or *picture* that uses *inter prediction*. **3.64** **inter prediction**: A *prediction* derived in a manner that is dependent on data elements (e.g., sample values or motion vectors) of one or more *reference pictures*. NOTE – A prediction from a reference picture that is the current picture itself is also inter prediction. **3.65** **intra coding**: Coding of a *coding block, slice*, or *picture* that uses *intra prediction*. **3.66** **intra prediction**: A *prediction* derived from only data elements (e.g., sample values) of the same decoded *slice* without referring to a *reference picture*. **3.69** **intra (I) slice**: A *slice* that is decoded using *intra prediction* only. **3.76** **list 0 (list 1) motion vector**: A *motion vector* associated with a *reference index* pointing into *reference picture list 0* (*list 1*). **3.77** **list 0 (list 1) prediction**: *Inter prediction* of the content of a *slice* using a *reference index* pointing into *reference picture list 0* (*list 1*). **3.82** **motion vector**: A two-dimensional vector used for *inter prediction* that provides an offset from the coordinates in the *decoded picture* to the coordinates in a *reference picture*.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	3.100 **prediction**: An embodiment of the *prediction process*. **3.101** **prediction block**: A rectangular MxN *block* of samples on which the same *prediction* is applied. **3.102** **prediction process**: The use of a *predictor* to provide an estimate of the data element (e.g., sample value or motion vector) currently being decoded. **3.103** **prediction unit**: A *prediction block* of *luma* samples, two corresponding *prediction blocks* of *chroma* samples of a *picture* that has three sample arrays, or a *prediction block* of samples of a monochrome *picture* or a *picture* that is coded using three separate colour planes and *syntax structures* used to predict the *prediction block* samples. **3.104** **predictive (P) slice**: A *slice* that is decoded using *intra prediction* or using *inter prediction* with at most one *motion vector* and *reference index* to *predict* the sample values of each *block*. **3.121** **reference index**: An index into a *reference picture list*. **3.122** **reference picture**: A *picture* that is a *short-term reference picture* or a *long-term reference picture*. NOTE – A reference picture contains samples that may be used for inter prediction in the decoding process of subsequent pictures in decoding order. **3.123** **reference picture list**: A list of *reference pictures* that is used for *inter prediction* of a *P* or *B slice*. NOTE – For the decoding process of a P slice, there is one reference picture list – reference picture list 0. For the decoding process of a B slice, there are two reference picture lists – reference picture list 0 and reference picture list 1. **3.124** **reference picture list 0**: The *reference picture list* used for *inter prediction* of a *P* or the first *reference picture list* used for *inter prediction* of a *B slice*. **3.125** **reference picture list 1**: The second *reference picture list* used for *inter prediction* of a *B slice*. **3.136** **slice**: An integer number of *coding tree units* contained in one *independent slice segment* and all subsequent *dependent slice segments* (if any) that precede the next *independent slice segment* (if any) within the same *access unit*. **3.153** **syntax element**: An element of data represented in the *bitstream*. **3.154** **syntax structure**: Zero or more *syntax elements* present together in the *bitstream* in a specified order. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 4-12.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	5.10 Variables, syntax elements and tables Syntax elements in the bitstream are represented in **bold** type. Each syntax element is described by its name (all lower case letters with underscore characters), and one descriptor for its method of coded representation. The decoding process behaves according to the value of the syntax element and to the values of previously decoded syntax elements. When a value of a syntax element is used in the syntax tables or the text, it appears in regular (i.e., not bold) type. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 18. **6.3 Partitioning of pictures, slices, slice segments, tiles, coding tree units and coding tree blocks** **6.3.1 Partitioning of pictures into slices, slice segments and tiles** This clause specifies how a picture is partitioned into slices, slice segments and tiles. Pictures are divided into slices and tiles. A slice is a sequence of one or more slice segments starting with an independent slice segment and containing all subsequent dependent slice segments (if any) that precede the next independent slice segment (if any) within the same picture. A slice segment is a sequence of coding tree units. Likewise, a tile is a sequence of coding tree units. For example, a picture may be divided into two slices as shown in Figure 6-4. In this example, the first slice is composed of an independent slice segment containing 4 coding tree units, a dependent slice segment containing 32 coding tree units and another dependent slice segment containing 24 coding tree units; and the second slice consists of a single independent slice segment containing the remaining 39 coding tree units of the picture. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 23. **7 Syntax and semantics** **7.1 Method of specifying syntax in tabular form** The syntax tables specify a superset of the syntax of all allowed bitstreams. Additional constraints on the syntax may be specified, either directly or indirectly, in other clauses. NOTE – An actual decoder should implement some means for identifying entry points into the bitstream and some means to identify and handle non-conforming bitstreams. The methods for identifying and handling errors and other such situations are not specified in this Specification. The following table lists examples of the syntax specification format. When **syntax_element** appears, it specifies that a syntax element is parsed from the bitstream and the bitstream pointer is advanced to the next position beyond the syntax element in the bitstream parsing process.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 30.

7.3.6 Slice segment header syntax

7.3.6.1 General slice segment header syntax

slice_segment_header( ) {	**Descriptor**
first_slice_segment_in_pic_flag	u(1)
if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
no_output_of_prior_pics_flag	u(1)
slice_pic_parameter_set_id	ue(v)
if(!first_slice_segment_in_pic_flag) {	
if(dependent_slice_segments_enabled_flag)	
dependent_slice_segment_flag	u(1)
slice_segment_address	u(v)
}	
if(!dependent_slice_segment_flag) {	
for(i = 0; i < num_extra_slice_header_bits; i++)	
slice_reserved_flag[i]	u(1)
slice_type	ue(v)
if(output_flag_present_flag)	

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 44.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	7.3.8.5 Coding unit syntax

<table>
<tr><td>coding_unit(x0, y0, log2CbSize) {</td><td>Descriptor</td></tr>
<tr><td>if(transquant_bypass_enabled_flag)</td><td></td></tr>
<tr><td>cu_transquant_bypass_flag</td><td>ae(v)</td></tr>
<tr><td>if(slice_type != I)</td><td></td></tr>
<tr><td>cu_skip_flag[x0][y0]</td><td>ae(v)</td></tr>
<tr><td>nCbS = (1 << log2CbSize)</td><td></td></tr>
<tr><td>if(cu_skip_flag[x0][y0])</td><td></td></tr>
<tr><td>prediction_unit(x0, y0, nCbS, nCbS)</td><td></td></tr>
<tr><td>else {</td><td></td></tr>
<tr><td>if(slice_type != I)</td><td></td></tr>
<tr><td>pred_mode_flag</td><td>ae(v)</td></tr>
<tr><td>if(palette_mode_enabled_flag && CuPredMode[x0][y0] = = MODE_INTRA &&
log2CbSize <= MaxTbLog2SizeY)</td><td></td></tr>
<tr><td>palette_mode_flag[x0][y0]</td><td>ae(v)</td></tr>
<tr><td>if(palette_mode_flag[x0][y0])</td><td></td></tr>
<tr><td>palette_coding(x0, y0, nCbS)</td><td></td></tr>
<tr><td>else {</td><td></td></tr>
<tr><td>if(CuPredMode[x0][y0] != MODE_INTRA ||
log2CbSize = = MinCbLog2SizeY)</td><td></td></tr>
</table>

<table>
<tr><td>part_mode</td><td>ae(v)</td></tr>
<tr><td>if(CuPredMode[x0][y0] = = MODE_INTRA) {</td><td></td></tr>
</table>

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 52-53.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	7.3.8.6 Prediction unit syntax (see syntax table below)

prediction_unit(x0, y0, nPbW, nPbH) {	**Descriptor**
if(cu_skip_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else { /* MODE_INTER */	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(slice_type = = B)	
inter_pred_idc[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0)	
mvp_l0_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] = = PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
} else	
mvd_coding(x0, y0, 1)	
mvp_l1_flag[x0][y0]	ae(v)
}	
}	
}	
}	

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 55.

slice_type specifies the coding type of the slice according to Table 7-7.

Table 7-7 – Name association to slice_type
<table>
<tr><th>slice_type</th><th>Name of slice_type</th></tr>
<tr><td>0</td><td>B (B slice)</td></tr>
<tr><td>1</td><td>P (P slice)</td></tr>
<tr><td>2</td><td>I (I slice)</td></tr>
</table>

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 95.

pred_mode_flag equal to 0 specifies that the current coding unit is coded in inter prediction mode. pred_mode_flag equal to 1 specifies that the current coding unit is coded in intra prediction mode. The variable CuPredMode[x][y] is derived as follows for x = x0..x0 + nCbS − 1 and y = y0..y0 + nCbS − 1:
– If pred_mode_flag is equal to 0, CuPredMode[x][y] is set equal to MODE_INTER.
– Otherwise (pred_mode_flag is equal to 1), CuPredMode[x][y] is set equal to MODE_INTRA.
When pred_mode_flag is not present, the variable CuPredMode[x][y] is derived as follows for x = x0..x0 + nCbS − 1 and y = y0..y0 + nCbS − 1:
– If slice_type is equal to I, CuPredMode[x][y] is inferred to be equal to MODE_INTRA.
– Otherwise (slice_type is equal to P or B), when cu_skip_flag[x0][y0] is equal to 1, CuPredMode[x][y] is inferred to be equal to MODE_SKIP.
</td></tr>
</table>

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS

	part_mode specifies the partitioning mode of the current coding unit. The semantics of part_mode depend on CuPredMode[x0][y0]. The variables PartMode and IntraSplitFlag are derived from the value of part_mode as defined in Table 7-10.
	
	The value of part_mode is restricted as follows:
	
	– If CuPredMode[x0][y0] is equal to MODE_INTRA, part_mode shall be equal to 0 or 1.
	
	– Otherwise (CuPredMode[x0][y0] is equal to MODE_INTER), the following applies:
	
	    – If log2CbSize is greater than MinCbLog2SizeY and amp_enabled_flag is equal to 1, part_mode shall be in the range of 0 to 2, inclusive, or in the range of 4 to 7, inclusive.
	
	    – Otherwise, if log2CbSize is greater than MinCbLog2SizeY and amp_enabled_flag is equal to 0, or log2CbSize is equal to 3, part_mode shall be in the range of 0 to 2, inclusive.
	
	    – Otherwise (log2CbSize is greater than 3 and equal to MinCbLog2SizeY), the value of part_mode shall be in the range of 0 to 3, inclusive.
	
	When part_mode is not present, the variables PartMode and IntraSplitFlag are derived as follows:
	
	– PartMode is set equal to PART_2Nx2N.
	
	– IntraSplitFlag is set equal to 0.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	Table 7-10 – Name association to prediction mode and partitioning type CuPredMode[x0][y0] / part_mode / IntraSplitFlag / PartMode: MODE_INTRA / 0 / 0 / PART_2Nx2N MODE_INTRA / 1 / 1 / PART_NxN MODE_INTER / 0 / 0 / PART_2Nx2N MODE_INTER / 1 / 0 / PART_2NxN MODE_INTER / 2 / 0 / PART_Nx2N MODE_INTER / 3 / 0 / PART_NxN MODE_INTER / 4 / 0 / PART_2NxnU MODE_INTER / 5 / 0 / PART_2NxnD MODE_INTER / 6 / 0 / PART_nLx2N MODE_INTER / 7 / 0 / PART_nRx2N **inter_pred_idc**[x0][y0] specifies whether list0, list1, or bi-prediction is used for the current prediction unit according to Table 7-11. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture. **Table 7-11 – Name association to inter prediction mode** inter_pred_idc / Name of inter_pred_idc (nPbW + nPbH) != 12 / Name of inter_pred_idc (nPbW + nPbH) == 12: 0 / PRED_L0 / PRED_L0 1 / PRED_L1 / PRED_L1 2 / PRED_BI / na

Table 7-10 – Name association to prediction mode and partitioning type

CuPredMode[x0][y0]	part_mode	IntraSplitFlag	PartMode
MODE_INTRA	0	0	PART_2Nx2N
	1	1	PART_NxN
MODE_INTER	0	0	PART_2Nx2N
	1	0	PART_2NxN
	2	0	PART_Nx2N
	3	0	PART_NxN
	4	0	PART_2NxnU
	5	0	PART_2NxnD
	6	0	PART_nLx2N
	7	0	PART_nRx2N

Table 7-11 – Name association to inter prediction mode

inter_pred_idc	Name of inter_pred_idc	
	(nPbW + nPbH) != 12	(nPbW + nPbH) == 12
0	PRED_L0	PRED_L0
1	PRED_L1	PRED_L1
2	PRED_BI	na

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 106-108. **6.2 Source, decoded and output picture formats** This clause specifies the relationship between source and decoded pictures that is given via the bitstream. The video source that is represented by the bitstream is a sequence of pictures in decoding order. The source and decoded pictures are each comprised of one or more sample arrays: – Luma (Y) only (monochrome). – Luma and two chroma (YCbCr or YCgCo). – Green, Blue and Red (GBR, also known as RGB). – Arrays representing other unspecified monochrome or tri-stimulus colour samplings (for example, YZX, also known as XYZ). For convenience of notation and terminology in this Specification, the variables and terms associated with these arrays are referred to as luma (or L or Y) and chroma, where the two chroma arrays are referred to as Cb and Cr; regardless of the actual colour representation method in use. The actual colour representation method in use can be indicated in syntax that is specified in Annex E. The number of bits necessary for the representation of each of the samples in the luma and chroma arrays in a video sequence is in the range of 8 to 16, inclusive, and the number of bits used in the luma array may differ from the number of bits used in the chroma arrays. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 21.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

7.3.2.2.1 General sequence parameter set RBSP syntax

seq_parameter_set_rbsp( ) {	**Descriptor**
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(1, sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc = = 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 34.

</td></tr>
</table>

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	bit_depth_luma_minus8 specifies the bit depth of the samples of the luma array $BitDepth_Y$ and the value of the luma quantization parameter range offset $QpBdOffset_Y$ as follows: $BitDepth_Y = 8 + bit\_depth\_luma\_minus8$ (7-4) $QpBdOffset_Y = 6 * bit\_depth\_luma\_minus8$ (7-5) bit_depth_luma_minus8 shall be in the range of 0 to 8, inclusive. **bit_depth_chroma_minus8** specifies the bit depth of the samples of the chroma arrays $BitDepth_C$ and the value of the chroma quantization parameter range offset $QpBdOffset_C$ as follows: $BitDepth_C = 8 + bit\_depth\_chroma\_minus8$ (7-6) $QpBdOffset_C = 6 * bit\_depth\_chroma\_minus8$ (7-7) bit_depth_chroma_minus8 shall be in the range of 0 to 8, inclusive. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 76.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3 Decoding process for prediction units in inter prediction mode **8.5.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable nCbS specifying the size of the current luma coding block, – a variable nPbW specifying the width of the current luma prediction block, – a variable nPbH specifying the height of the current luma prediction block, – a variable partIdx specifying the index of the current prediction unit within the current coding unit. Outputs of this process are: – an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of luma prediction samples, where $nCbS_L$ is derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below. The decoding process for prediction units in inter prediction mode consists of the following ordered steps: 1. The derivation process for motion vector components and reference indices as specified in clause 8.5.3.2 is invoked with the luma coding block location (xCb, yCb), the luma prediction block location (xBl, yBl), the luma coding block size block nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH and the prediction unit index partIdx as inputs, and the luma motion vectors mvL0 and mvL1, when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1 and the prediction list utilization flags predFlagL0 and predFlagL1 as outputs. 2. The decoding process for inter sample prediction as specified in clause 8.5.3.3 is invoked with the luma coding block location (xCb, yCb), the luma prediction block location (xBl, yBl), the luma coding block size block nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH, the luma motion vectors mvL0 and mvL1, when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, the

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	reference indices refIdxL0 and refIdxL1, and the prediction list utilization flags predFlagL0 and predFlagL1 as inputs, and the inter prediction samples (predSamples) that are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of prediction luma samples and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cr}$ and $predSamples_{Cr}$ of prediction chroma samples, one for each of the chroma components Cb and Cr, as outputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 144-45. **8.5.3.2 Derivation process for motion vector components and reference indices** **8.5.3.2.1 General** Inputs to this process are: – a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (xBl, yBl) of the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable nCbS specifying the size of the current luma coding block, – two variables nPbW and nPbH specifying the width and the height of the luma prediction block, – a variable partIdx specifying the index of the current prediction unit within the current coding unit. Outputs of this process are: – the luma motion vectors mvL0 and mvL1, – when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, – the reference indices refIdxL0 and refIdxL1, – the prediction list utilization flags predFlagL0 and predFlagL1. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 144-45.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	For the derivation of the variables mvL0 and mvL1, refIdxL0 and refIdxL1, as well as predFlagL0 and predFlagL1, the following applies: – If merge_flag[xPb][yPb] is equal to 1, the derivation process for luma motion vectors for merge mode as specified in clause 8.5.3.2.2 is invoked with the luma location (xCb, yCb), the luma location (xPb, yPb), the variables nCbS, nPbW, nPbH and the partition index partIdx as inputs, and the output being the luma motion vectors mvL0, mvL1, the reference indices refIdxL0, refIdxL1 and the prediction list utilization flags predFlagL0 and predFlagL1. – Otherwise, for X being replaced by either 0 or 1 in the variables predFlagLX, mvLX and refIdxLX, in PRED_LX, and in the syntax elements ref_idx_lX and MvdLX, the following applies: 1. The variables refIdxLX and predFlagLX are derived as follows: – If inter_pred_idc[xPb][yPb] is equal to PRED_LX or PRED_BI, refIdxLX = ref_idx_lX[xPb][yPb] (8-88) predFlagLX = 1 (8-89) – Otherwise, the variables refIdxLX and predFlagLX are specified by: refIdxLX = −1 (8-90) predFlagLX = 0 (8-91) 2. The variable mvdLX is derived as follows: mvdLX[0] = MvdLX[xPb][yPb][0] (8-92) mvdLX[1] = MvdLX[xPb][yPb][1] (8-93)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	4. When predFlagLX is equal to 1 and the picture with index refIdx from reference picture list LX of the slice is not the current picture, and use_integer_mv_flag is equal to 0, the luma motion vector mvLX is derived as follows: $uLX[0] = (mvpLX[0] + mvdLX[0] + 2^{16}) \% 2^{16}$ (8-94) $mvLX[0] = (uLX[0] >= 2^{15}) ? (uLX[0] - 2^{16}) : uLX[0]$ (8-95) $uLX[1] = (mvpLX[1] + mvdLX[1] + 2^{16}) \% 2^{16}$ (8-96) $mvLX[1] = (uLX[1] >= 2^{15}) ? (uLX[1] - 2^{16}) : uLX[1]$ (8-97) NOTE 1– The resulting values of mvLX[0] and mvLX[1] as specified above will always be in the range of -2^{15} to $2^{15} - 1$, inclusive. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 145-46. **8.5.3.3 Decoding process for inter prediction samples** **8.5.3.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable nCbS specifying the size of the current luma coding block, – two variables nPbW and nPbH specifying the width and the height of the luma prediction block, – the luma motion vectors mvL0 and mvL1,

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, – the reference indices refIdxL0 and refIdxL1, – the prediction list utilization flags, predFlagL0, and predFlagL1. Outputs of this process are: – an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of luma prediction samples, where $nCbS_L$ is derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below. The variable $nCbS_L$ is set equal to nCbS. When ChromaArrayType is not equal to 0, the variable $nCbSw_C$ is set equal to nCbS / SubWidthC and the variable $nCbSh_C$ is set equal to nCbS / SubHeightC. Let $predSamplesL0_L$ and $predSamplesL1_L$ be (nPbW)x(nPbH) arrays of predicted luma sample values and, when ChromaArrayType is not equal to 0, $predSamplesL0_{Cb}$, $predSamplesL1_{Cb}$, $predSamplesL0_{Cr}$ and $predSamplesL1_{Cr}$ be (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays of predicted chroma sample values. For X being each of 0 and 1, when predFlagLX is equal to 1, the following applies: – The reference picture consisting of an ordered two-dimensional array $refPicLX_L$ of luma samples and, when ChromaArrayType is not equal to 0, two ordered two-dimensional arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$ of chroma samples is derived by invoking the process specified in clause 8.5.3.3.2 with refIdxLX as input. – The array $predSamplesLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ are derived by invoking the fractional sample interpolation process specified in clause 8.5.3.3.3 with the luma locations (xCb, yCb) and (xBl, yBl), the luma prediction block width nPbW, the luma prediction block height nPbH, the motion vectors mvLX and, when ChromaArrayType is not equal to 0, mvCLX, and the reference arrays $refPicLX_L$, $refPicLX_{Cb}$, and $refPicLX_{Cr}$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 161-62.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3.3 Fractional sample interpolation process **8.5.3.3.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block – two variables nPbW and nPbH specifying the width and the height of the luma prediction block – a luma motion vector mvLX given in quarter-luma-sample units – when ChromaArrayType is not equal to 0, a chroma motion vector mvCLX given in eighth-chroma-sample units – the selected reference picture sample array $refPicLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$. Outputs of this process are: – an (nPbW)x(nPbH) array $predSamplesLX_L$ of prediction luma sample values – when ChromaArrayType is not equal to 0, two (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ of prediction chroma sample values. The location (xPb, yPb) given in full-sample units of the upper-left luma samples of the current prediction block relative to the upper-left luma sample location of the given reference sample arrays is derived as follows: $xPb = xCb + xBl$ (8-214) $yPb = yCb + yBl$ (8-215)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	Let ($xInt_L$, $yInt_L$) be a luma location given in full-sample units and ($xFrac_L$, $yFrac_L$) be an offset given in quarter-sample units. These variables are used only inside this clause for specifying fractional-sample locations inside the reference sample arrays $refPicLX_L$, $refPicLX_{Cb}$ and $refPicLX_{Cr}$. For each luma sample location ($x_L = 0..nPbW - 1$, $y_L = 0..nPbH - 1$) inside the prediction luma sample array $predSamplesLX_L$, the corresponding prediction luma sample value $predSamplesLX_L[x_L][y_L]$ is derived as follows: – The variables $xInt_L$, $yInt_L$, $xFrac_L$ and $yFrac_L$ are derived as follows: $xInt_L = xPb + (mvLX[0] >> 2) + x_L$ (8-216) $yInt_L = yPb + (mvLX[1] >> 2) + y_L$ (8-217) $xFrac_L = mvLX[0] \& 3$ (8-218) $yFrac_L = mvLX[1] \& 3$ (8-219) – The prediction luma sample value $predSamplesLX_L[x_L][y_L]$ is derived by invoking the process specified in clause 8.5.3.3.3.2 with ($xInt_L$, $yInt_L$), ($xFrac_L$, $yFrac_L$) and $refPicLX_L$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 163.
[C] using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;	Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding video comprising using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision. Each of the Accused Products performs a method for decoding video comprising using said first reference block to obtain a first prediction (for example, through the processes for obtaining $predSamplesLX_L$, $predSamplesLX_C$ shown below), said first prediction having a second precision, which is higher than said first precision. The following specifications provide further evidence of how each of the Accused Products operates:

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3 Decoding process for inter prediction samples **8.5.3.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable nCbS specifying the size of the current luma coding block, – two variables nPbW and nPbH specifying the width and the height of the luma prediction block, – the luma motion vectors mvL0 and mvL1,

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, – the reference indices refIdxL0 and refIdxL1, – the prediction list utilization flags, predFlagL0, and predFlagL1. Outputs of this process are: – an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of luma prediction samples, where $nCbS_L$ is derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below. The variable $nCbS_L$ is set equal to nCbS. When ChromaArrayType is not equal to 0, the variable $nCbSw_C$ is set equal to nCbS / SubWidthC and the variable $nCbSh_C$ is set equal to nCbS / SubHeightC. Let $predSamplesL0_L$ and $predSamplesL1_L$ be (nPbW)x(nPbH) arrays of predicted luma sample values and, when ChromaArrayType is not equal to 0, $predSamplesL0_{Cb}$, $predSamplesL1_{Cb}$, $predSamplesL0_{Cr}$ and $predSamplesL1_{Cr}$ be (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays of predicted chroma sample values. For X being each of 0 and 1, when predFlagLX is equal to 1, the following applies: – The reference picture consisting of an ordered two-dimensional array $refPicLX_L$ of luma samples and, when ChromaArrayType is not equal to 0, two ordered two-dimensional arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$ of chroma samples is derived by invoking the process specified in clause 8.5.3.3.2 with refIdxLX as input. – The array $predSamplesLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ are derived by invoking the fractional sample interpolation process specified in clause 8.5.3.3.3 with the luma locations (xCb, yCb) and (xBl, yBl), the luma prediction block width nPbW, the luma prediction block height nPbH, the motion vectors mvLX and, when ChromaArrayType is not equal to 0, mvCLX, and the reference arrays $refPicLX_L$, $refPicLX_{Cb}$, and $refPicLX_{Cr}$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 161-62.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3.3 Fractional sample interpolation process **8.5.3.3.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block – two variables nPbW and nPbH specifying the width and the height of the luma prediction block – a luma motion vector mvLX given in quarter-luma-sample units – when ChromaArrayType is not equal to 0, a chroma motion vector mvCLX given in eighth-chroma-sample units – the selected reference picture sample array $refPicLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$. Outputs of this process are: – an (nPbW)x(nPbH) array $predSamplesLX_L$ of prediction luma sample values – when ChromaArrayType is not equal to 0, two (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ of prediction chroma sample values. The location (xPb, yPb) given in full-sample units of the upper-left luma samples of the current prediction block relative to the upper-left luma sample location of the given reference sample arrays is derived as follows: $xPb = xCb + xBl$ (8-214) $yPb = yCb + yBl$ (8-215)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	Let ($xInt_L$, $yInt_L$) be a luma location given in full-sample units and ($xFrac_L$, $yFrac_L$) be an offset given in quarter-sample units. These variables are used only inside this clause for specifying fractional-sample locations inside the reference sample arrays $refPicLX_L$, $refPicLX_{Cb}$ and $refPicLX_{Cr}$. For each luma sample location ($x_L = 0..nPbW - 1$, $y_L = 0..nPbH - 1$) inside the prediction luma sample array $predSamplesLX_L$, the corresponding prediction luma sample value $predSamplesLX_L[x_L][y_L]$ is derived as follows: – The variables $xInt_L$, $yInt_L$, $xFrac_L$ and $yFrac_L$ are derived as follows: $xInt_L = xPb + (mvLX[0] >> 2) + x_L$ (8-216) $yInt_L = yPb + (mvLX[1] >> 2) + y_L$ (8-217) $xFrac_L = mvLX[0] \& 3$ (8-218) $yFrac_L = mvLX[1] \& 3$ (8-219) – The prediction luma sample value $predSamplesLX_L[x_L][y_L]$ is derived by invoking the process specified in clause 8.5.3.3.3.2 with ($xInt_L$, $yInt_L$), ($xFrac_L$, $yFrac_L$) and $refPicLX_L$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 163. **8.5.3.3.3.2 Luma sample interpolation process** Inputs to this process are: – a luma location in full-sample units ($xInt_L$, $yInt_L$), – a luma location in fractional-sample units ($xFrac_L$, $yFrac_L$), – the luma reference sample array $refPicLX_L$. Output of this process is a predicted luma sample value $predSampleLX_L$

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS

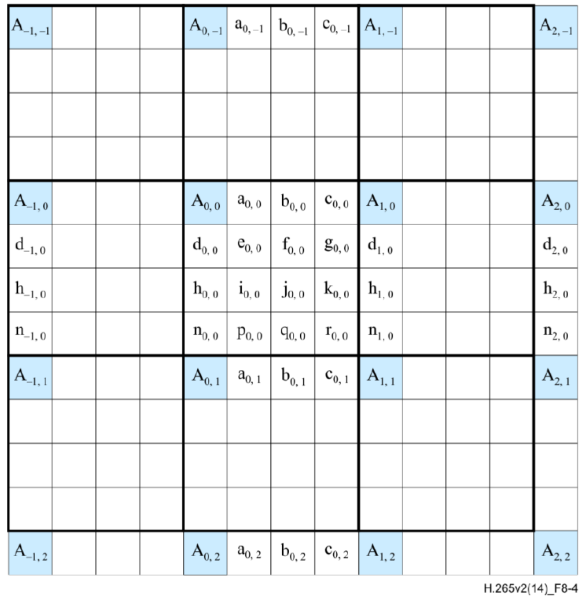


Figure 8-4 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation

In Figure 8-4, the positions labelled with upper-case letters $A_{i,j}$ within shaded blocks represent luma samples at full-sample locations inside the given two-dimensional array $refPicLX_L$ of luma samples. These samples may be used for generating the predicted luma sample value $predSampleLX_L$. The locations ($xA_{i,j}$, $yA_{i,j}$) for each of the corresponding luma samples $A_{i,j}$ inside the given array $refPicLX_L$ of luma samples are derived as follows:

$$xA_{i,j} = \text{Clip3}(0, \text{pic\_width\_in\_luma\_samples} - 1, xInt_L + i) \quad (8\text{-}224)$$

$$yA_{i,j} = \text{Clip3}(0, \text{pic\_height\_in\_luma\_samples} - 1, yInt_L + j) \quad (8\text{-}225)$$

The positions labelled with lower-case letters within un-shaded blocks represent luma samples at quarter-luma-sample fractional locations. The luma location offset in fractional-sample units ($xFrac_L$, $yFrac_L$) specifies which of the generated

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	luma samples at full-sample and fractional-sample locations is assigned to the predicted luma sample value $predSampleLX_L$. This assignment is as specified in Table 8-8. The value of $predSampleLX_L$ is the output. The variables shift1, shift2 and shift3 are derived as follows: – The variable shift1 is set equal to $Min(4, BitDepth_Y - 8)$, the variable shift2 is set equal to 6 and the variable shift3 is set equal to $Max(2, 14 - BitDepth_Y)$. Given the luma samples $A_{i,j}$ at full-sample locations $(xA_{i,j}, yA_{i,j})$, the luma samples $a_{0,0}$ to $r_{0,0}$ at fractional sample positions are derived as follows: – The samples labelled $a_{0,0}$, $b_{0,0}$, $c_{0,0}$, $d_{0,0}$, $h_{0,0}$ and $n_{0,0}$ are derived by applying an 8-tap filter to the nearest integer position samples as follows: $a_{0,0} = (-A_{-3,0} + 4 * A_{-2,0} - 10 * A_{-1,0} + 58 * A_{0,0} + 17 * A_{1,0} - 5 * A_{2,0} + A_{3,0}) >> shift1$ (8-226) $b_{0,0} = (-A_{-3,0} + 4 * A_{-2,0} - 11 * A_{-1,0} + 40 * A_{0,0} + 40 * A_{1,0} - 11 * A_{2,0} + 4 * A_{3,0} - A_{4,0}) >> shift1$ (8-227) $c_{0,0} = (A_{-2,0} - 5 * A_{-1,0} + 17 * A_{0,0} + 58 * A_{1,0} - 10 * A_{2,0} + 4 * A_{3,0} - A_{4,0}) >> shift1$ (8-228) $d_{0,0} = (-A_{0,-3} + 4 * A_{0,-2} - 10 * A_{0,-1} + 58 * A_{0,0} + 17 * A_{0,1} - 5 * A_{0,2} + A_{0,3}) >> shift1$ (8-229) $h_{0,0} = (-A_{0,-3} + 4 * A_{0,-2} - 11 * A_{0,-1} + 40 * A_{0,0} + 40 * A_{0,1} - 11 * A_{0,2} + 4 * A_{0,3} - A_{0,4}) >> shift1$ (8-230) $n_{0,0} = (A_{0,-2} - 5 * A_{0,-1} + 17 * A_{0,0} + 58 * A_{0,1} - 10 * A_{0,2} + 4 * A_{0,3} - A_{0,4}) >> shift1$ (8-231)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

– The samples labelled $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $j_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$ and $r_{0,0}$ are derived by applying an 8-tap filter to the samples $a_{0,i}$, $b_{0,i}$ and $c_{0,i}$ with $i = -3..4$ in the vertical direction as follows:

$$e_{0,0} = (-a_{0,-3} + 4 * a_{0,-2} - 10 * a_{0,-1} + 58 * a_{0,0} + 17 * a_{0,1} - 5 * a_{0,2} + a_{0,3}) >> \text{shift2} \quad (8\text{-}232)$$

$$i_{0,0} = (-a_{0,-3} + 4 * a_{0,-2} - 11 * a_{0,-1} + 40 * a_{0,0} + 40 * a_{0,1} - 11 * a_{0,2} + 4 * a_{0,3} - a_{0,4}) >> \text{shift2} \quad (8\text{-}233)$$

$$p_{0,0} = (a_{0,-2} - 5 * a_{0,-1} + 17 * a_{0,0} + 58 * a_{0,1} - 10 * a_{0,2} + 4 * a_{0,3} - a_{0,4}) >> \text{shift2} \quad (8\text{-}234)$$

$$f_{0,0} = (-b_{0,-3} + 4 * b_{0,-2} - 10 * b_{0,-1} + 58 * b_{0,0} + 17 * b_{0,1} - 5 * b_{0,2} + b_{0,3}) >> \text{shift2} \quad (8\text{-}235)$$

$$j_{0,0} = (-b_{0,-3} + 4 * b_{0,-2} - 11 * b_{0,-1} + 40 * b_{0,0} + 40 * b_{0,1} - 11 * b_{0,2} + 4 * b_{0,3} - b_{0,4}) >> \text{shift2} \quad (8\text{-}236)$$

$$q_{0,0} = (b_{0,-2} - 5 * b_{0,-1} + 17 * b_{0,0} + 58 * b_{0,1} - 10 * b_{0,2} + 4 * b_{0,3} - b_{0,4}) >> \text{shift2} \quad (8\text{-}237)$$

$$g_{0,0} = (-c_{0,-3} + 4 * c_{0,-2} - 10 * c_{0,-1} + 58 * c_{0,0} + 17 * c_{0,1} - 5 * c_{0,2} + c_{0,3}) >> \text{shift2} \quad (8\text{-}238)$$

$$k_{0,0} = (-c_{0,-3} + 4 * c_{0,-2} - 11 * c_{0,-1} + 40 * c_{0,0} + 40 * c_{0,1} - 11 * c_{0,2} + 4 * c_{0,3} - c_{0,4}) >> \text{shift2} \quad (8\text{-}239)$$

$$r_{0,0} = (c_{0,-2} - 5 * c_{0,-1} + 17 * c_{0,0} + 58 * c_{0,1} - 10 * c_{0,2} + 4 * c_{0,3} - c_{0,4}) >> \text{shift2} \quad (8\text{-}240)$$

Table 8-8 – Assignment of the luma prediction sample predSampleLX$_L$

xFracL	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
yFracL	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
predSampleLX$_L$	A << shift3	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 164-65.

8.5.3.3.3.3 Chroma sample interpolation process

This process is only invoked when ChromaArrayType is not equal to 0.

Inputs to this process are:

– a chroma location in full-sample units ($xInt_C$, $yInt_C$),

</td></tr>
</table>

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– a chroma location in eighth fractional-sample units ($xFrac_C$, $yFrac_C$), – the chroma reference sample array $refPicLX_C$. Output of this process is a predicted chroma sample value $predSampleLX_C$

	$ha_{0,-1}$	$hb_{0,-1}$	$hc_{0,-1}$	$hd_{0,-1}$	$he_{0,-1}$	$hf_{0,-1}$	$hg_{0,-1}$	$hh_{0,-1}$	
$ah_{-1,0}$	$B_{0,0}$	$ab_{0,0}$	$ac_{0,0}$	$ad_{0,0}$	$ae_{0,0}$	$af_{0,0}$	$ag_{0,0}$	$ah_{0,0}$	$B_{1,0}$
$bh_{-1,0}$	$ba_{0,0}$	$bb_{0,0}$	$bc_{0,0}$	$bd_{0,0}$	$be_{0,0}$	$bf_{0,0}$	$bg_{0,0}$	$bh_{0,0}$	$ba_{1,0}$
$ch_{-1,0}$	$ca_{0,0}$	$cb_{0,0}$	$cc_{0,0}$	$cd_{0,0}$	$ce_{0,0}$	$cf_{0,0}$	$cg_{0,0}$	$ch_{0,0}$	$ca_{1,0}$
$dh_{-1,0}$	$da_{0,0}$	$db_{0,0}$	$dc_{0,0}$	$dd_{0,0}$	$de_{0,0}$	$df_{0,0}$	$dg_{0,0}$	$dh_{0,0}$	$da_{1,0}$
$eh_{-1,0}$	$ea_{0,0}$	$eb_{0,0}$	$ec_{0,0}$	$ed_{0,0}$	$ee_{0,0}$	$ef_{0,0}$	$eg_{0,0}$	$eh_{0,0}$	$ea_{1,0}$
$fh_{-1,0}$	$fa_{0,0}$	$fb_{0,0}$	$fc_{0,0}$	$fd_{0,0}$	$fe_{0,0}$	$ff_{0,0}$	$fg_{0,0}$	$fh_{0,0}$	$fa_{1,0}$
$gh_{-1,0}$	$ga_{0,0}$	$gb_{0,0}$	$gc_{0,0}$	$gd_{0,0}$	$ge_{0,0}$	$gf_{0,0}$	$gg_{0,0}$	$gh_{0,0}$	$ga_{1,0}$
$hh_{-1,0}$	$ha_{0,0}$	$hb_{0,0}$	$hc_{0,0}$	$hd_{0,0}$	$he_{0,0}$	$hf_{0,0}$	$hg_{0,0}$	$hh_{0,0}$	$ha_{1,0}$
	$B_{0,1}$	$ab_{0,1}$	$ac_{0,1}$	$ad_{0,1}$	$ae_{0,1}$	$af_{0,1}$	$ag_{0,1}$	$ah_{0,1}$	$B_{1,1}$

H.265v2(14)_F8-5

Figure 8-5 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for eighth sample chroma interpolation

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	In Figure 8-5, the positions labelled with upper-case letters $B_{i,j}$ within shaded blocks represent chroma samples at full-sample locations inside the given two-dimensional array $refPicLX_C$ of chroma samples. These samples may be used for generating the predicted chroma sample value $predSampleLX_C$. The locations ($xB_{i,j}$, $yB_{i,j}$) for each of the corresponding chroma samples $B_{i,j}$ inside the given array $refPicLX_C$ of chroma samples are derived as follows: $xB_{i,j} = \text{Clip3}(0, (\text{pic\_width\_in\_luma\_samples} / \text{SubWidthC}) - 1, xInt_C + i)$ (8-241) $yB_{i,j} = \text{Clip3}(0, (\text{pic\_height\_in\_luma\_samples} / \text{SubHeightC}) - 1, yInt_C + j)$ (8-242) The positions labelled with lower-case letters within un-shaded blocks represent chroma samples at eighth-pel sample fractional locations. The chroma location offset in fractional-sample units ($xFrac_C$, $yFrac_C$) specifies which of the generated chroma samples at full-sample and fractional-sample locations is assigned to the predicted chroma sample value $predSampleLX_C$. This assignment is as specified in Table 8-9. The output is the value of $predSampleLX_C$. The variables shift1, shift2 and shift3 are derived as follows: – The variable shift1 is set equal to Min(4, $BitDepth_C - 8$), the variable shift2 is set equal to 6 and the variable shift3 is set equal to Max(2, $14 - BitDepth_C$). Given the chroma samples $B_{i,j}$ at full-sample locations ($xB_{i,j}$, $yB_{i,j}$), the chroma samples $ab_{0,0}$ to $hh_{0,0}$ at fractional sample positions are derived as follows: – The samples labelled $ab_{0,0}$, $ac_{0,0}$, $ad_{0,0}$, $ae_{0,0}$, $af_{0,0}$, $ag_{0,0}$ and $ah_{0,0}$ are derived by applying a 4-tap filter to the nearest integer position samples as follows: $ab_{0,0} = (-2 * B_{-1,0} + 58 * B_{0,0} + 10 * B_{1,0} - 2 * B_{2,0}) >> \text{shift1}$ (8-243) $ac_{0,0} = (-4 * B_{-1,0} + 54 * B_{0,0} + 16 * B_{1,0} - 2 * B_{2,0}) >> \text{shift1}$ (8-244) $ad_{0,0} = (-6 * B_{-1,0} + 46 * B_{0,0} + 28 * B_{1,0} - 4 * B_{2,0}) >> \text{shift1}$ (8-245) $ae_{0,0} = (-4 * B_{-1,0} + 36 * B_{0,0} + 36 * B_{1,0} - 4 * B_{2,0}) >> \text{shift1}$ (8-246) $af_{0,0} = (-4 * B_{-1,0} + 28 * B_{0,0} + 46 * B_{1,0} - 6 * B_{2,0}) >> \text{shift1}$ (8-247)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	$ag_{0,0} = (-2 * B_{-1,0} + 16 * B_{0,0} + 54 * B_{1,0} - 4 * B_{2,0}) >> shift1$ (8-248) $ah_{0,0} = (-2 * B_{-1,0} + 10 * B_{0,0} + 58 * B_{1,0} - 2 * B_{2,0}) >> shift1$ (8-249) – The samples labelled $ba_{0,0}$, $ca_{0,0}$, $da_{0,0}$, $ea_{0,0}$, $fa_{0,0}$, $ga_{0,0}$ and $ha_{0,0}$ are derived by applying a 4-tap filter to the nearest integer position samples as follows: $ba_{0,0} = (-2 * B_{0,-1} + 58 * B_{0,0} + 10 * B_{0,1} - 2 * B_{0,2}) >> shift1$ (8-250) $ca_{0,0} = (-4 * B_{0,-1} + 54 * B_{0,0} + 16 * B_{0,1} - 2 * B_{0,2}) >> shift1$ (8-251) $da_{0,0} = (-6 * B_{0,-1} + 46 * B_{0,0} + 28 * B_{0,1} - 4 * B_{0,2}) >> shift1$ (8-252) $ea_{0,0} = (-4 * B_{0,-1} + 36 * B_{0,0} + 36 * B_{0,1} - 4 * B_{0,2}) >> shift1$ (8-253) $fa_{0,0} = (-4 * B_{0,-1} + 28 * B_{0,0} + 46 * B_{0,1} - 6 * B_{0,2}) >> shift1$ (8-254) $ga_{0,0} = (-2 * B_{0,-1} + 16 * B_{0,0} + 54 * B_{0,1} - 4 * B_{0,2}) >> shift1$ (8-255) $ha_{0,0} = (-2 * B_{0,-1} + 10 * B_{0,0} + 58 * B_{0,1} - 2 * B_{0,2}) >> shift1$ (8-256) – The samples labelled $bX_{0,0}$, $cX_{0,0}$, $dX_{0,0}$, $eX_{0,0}$, $fX_{0,0}$, $gX_{0,0}$ and $hX_{0,0}$ for X being replaced by b, c, d, e, f, g and h, respectively, are derived by applying a 4-tap filter to the intermediate values $aX_{0,i}$ with $i = -1..2$ in the vertical direction as follows: $bX_{0,0} = (-2 * aX_{0,-1} + 58 * aX_{0,0} + 10 * aX_{0,1} - 2 * aX_{0,2}) >> shift2$ (8-257) $cX_{0,0} = (-4 * aX_{0,-1} + 54 * aX_{0,0} + 16 * aX_{0,1} - 2 * aX_{0,2}) >> shift2$ (8-258) $dX_{0,0} = (-6 * aX_{0,-1} + 46 * aX_{0,0} + 28 * aX_{0,1} - 4 * aX_{0,2}) >> shift2$ (8-259) $eX_{0,0} = (-4 * aX_{0,-1} + 36 * aX_{0,0} + 36 * aX_{0,1} - 4 * aX_{0,2}) >> shift2$ (8-260)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

$$fX_{0,0} = (-4 * aX_{0,-1} + 28 * aX_{0,0} + 46 * aX_{0,1} - 6 * aX_{0,2}) >> shift2 \quad (8\text{-}261)$$

$$gX_{0,0} = (-2 * aX_{0,-1} + 16 * aX_{0,0} + 54 * aX_{0,1} - 4 * aX_{0,2}) >> shift2 \quad (8\text{-}262)$$

$$hX_{0,0} = (-2 * aX_{0,-1} + 10 * aX_{0,0} + 58 * aX_{0,1} - 2 * aX_{0,2}) >> shift2 \quad (8\text{-}263)$$

Table 8-9 – Assignment of the chroma prediction sample predSampleLXC for (X, Y) being replaced by (1, b), (2, c), (3, d), (4, e), (5, f), (6, g) and (7, h), respectively

<table>
<tr><td>xFracC</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>yFracC</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>$predSampleLX_C$</td><td>B << shift3</td><td>ba</td><td>ca</td><td>da</td><td>ea</td><td>fa</td><td>ga</td><td>ha</td></tr>
<tr><td colspan="9"></td></tr>
<tr><td>xFracC</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>yFracC</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>$predSampleLX_C$</td><td>aY</td><td>bY</td><td>cY</td><td>dY</td><td>eY</td><td>fY</td><td>gY</td><td>hY</td></tr>
</table>

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 165-67.
</td></tr>
<tr><td>[D] using said second reference block to obtain a second prediction, said second prediction having the second precision;</td><td>Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding video comprising, using said second reference block to obtain a second prediction, said second prediction having the second precision.

Each of the Accused Products performs a method for decoding video comprising, using said second reference block to obtain a second prediction (for example, through the processes for obtaining $predSamplesLX_L$, $predSamplesLX_C$ shown below), said second prediction having the second precision, corresponding to the decoding process specified by the H.265 Standard. The following specifications provide further evidence of how each of the Accused Products operates:</td></tr>
</table>

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3 Decoding process for inter prediction samples **8.5.3.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable nCbS specifying the size of the current luma coding block, – two variables nPbW and nPbH specifying the width and the height of the luma prediction block, – the luma motion vectors mvL0 and mvL1,

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, – the reference indices refIdxL0 and refIdxL1, – the prediction list utilization flags, predFlagL0, and predFlagL1. Outputs of this process are: – an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of luma prediction samples, where $nCbS_L$ is derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below. The variable $nCbS_L$ is set equal to nCbS. When ChromaArrayType is not equal to 0, the variable $nCbSw_C$ is set equal to nCbS / SubWidthC and the variable $nCbSh_C$ is set equal to nCbS / SubHeightC. Let $predSamplesL0_L$ and $predSamplesL1_L$ be (nPbW)x(nPbH) arrays of predicted luma sample values and, when ChromaArrayType is not equal to 0, $predSamplesL0_{Cb}$, $predSamplesL1_{Cb}$, $predSamplesL0_{Cr}$ and $predSamplesL1_{Cr}$ be (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays of predicted chroma sample values. For X being each of 0 and 1, when predFlagLX is equal to 1, the following applies: – The reference picture consisting of an ordered two-dimensional array $refPicLX_L$ of luma samples and, when ChromaArrayType is not equal to 0, two ordered two-dimensional arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$ of chroma samples is derived by invoking the process specified in clause 8.5.3.3.2 with refIdxLX as input. – The array $predSamplesLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ are derived by invoking the fractional sample interpolation process specified in clause 8.5.3.3.3 with the luma locations (xCb, yCb) and (xBl, yBl), the luma prediction block width nPbW, the luma prediction block height nPbH, the motion vectors mvLX and, when ChromaArrayType is not equal to 0, mvCLX, and the reference arrays $refPicLX_L$, $refPicLX_{Cb}$, and $refPicLX_{Cr}$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 161-62.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

8.5.3.3.3 Fractional sample interpolation process

8.5.3.3.3.1 General

Inputs to this process are:

– a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture

– a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block

– two variables nPbW and nPbH specifying the width and the height of the luma prediction block

– a luma motion vector mvLX given in quarter-luma-sample units

– when ChromaArrayType is not equal to 0, a chroma motion vector mvCLX given in eighth-chroma-sample units

– the selected reference picture sample array $refPicLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$.

Outputs of this process are:

– an (nPbW)x(nPbH) array $predSamplesLX_L$ of prediction luma sample values

– when ChromaArrayType is not equal to 0, two (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ of prediction chroma sample values.

The location (xPb, yPb) given in full-sample units of the upper-left luma samples of the current prediction block relative to the upper-left luma sample location of the given reference sample arrays is derived as follows:

$$xPb = xCb + xBl \quad (8\text{-}214)$$

$$yPb = yCb + yBl \quad (8\text{-}215)$$

</td></tr>
</table>

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	Let ($xInt_L$, $yInt_L$) be a luma location given in full-sample units and ($xFrac_L$, $yFrac_L$) be an offset given in quarter-sample units. These variables are used only inside this clause for specifying fractional-sample locations inside the reference sample arrays $refPicLX_L$, $refPicLX_{Cb}$ and $refPicLX_{Cr}$. For each luma sample location ($x_L = 0..nPbW - 1$, $y_L = 0..nPbH - 1$) inside the prediction luma sample array $predSamplesLX_L$, the corresponding prediction luma sample value $predSamplesLX_L[x_L][y_L]$ is derived as follows: – The variables $xInt_L$, $yInt_L$, $xFrac_L$ and $yFrac_L$ are derived as follows: $xInt_L = xPb + (mvLX[0] >> 2) + x_L$ (8-216) $yInt_L = yPb + (mvLX[1] >> 2) + y_L$ (8-217) $xFrac_L = mvLX[0] \& 3$ (8-218) $yFrac_L = mvLX[1] \& 3$ (8-219) – The prediction luma sample value $predSamplesLX_L[x_L][y_L]$ is derived by invoking the process specified in clause 8.5.3.3.3.2 with ($xInt_L$, $yInt_L$), ($xFrac_L$, $yFrac_L$) and $refPicLX_L$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 163. **8.5.3.3.3.2 Luma sample interpolation process** Inputs to this process are: – a luma location in full-sample units ($xInt_L$, $yInt_L$), – a luma location in fractional-sample units ($xFrac_L$, $yFrac_L$), – the luma reference sample array $refPicLX_L$. Output of this process is a predicted luma sample value $predSampleLX_L$

EXHIBIT 10
United States Patent No. 11,805,267
Claim Chart for Infringement of Claim 19 by Hisense Accused Products

U.S. Patent No. 11,805,267	Hisense Accused Products

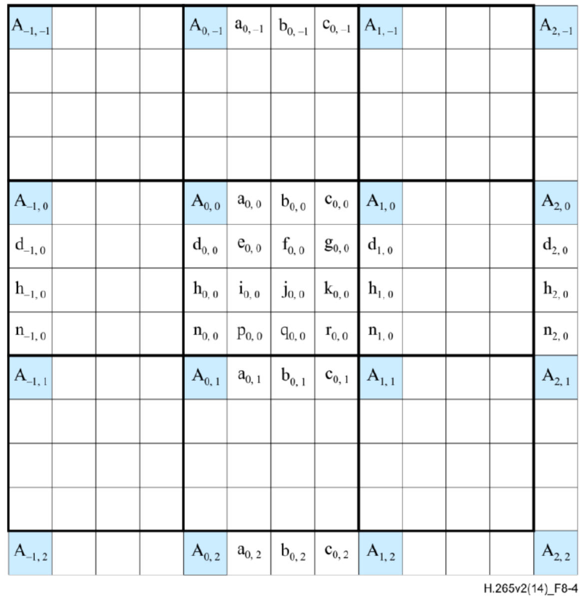


Figure 8-4 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation

In Figure 8-4, the positions labelled with upper-case letters $A_{i,j}$ within shaded blocks represent luma samples at full-sample locations inside the given two-dimensional array $refPicLX_L$ of luma samples. These samples may be used for generating the predicted luma sample value $predSampleLX_L$. The locations ($xA_{i,j}$, $yA_{i,j}$) for each of the corresponding luma samples $A_{i,j}$ inside the given array $refPicLX_L$ of luma samples are derived as follows:

$$xA_{i,j} = \text{Clip3}(\ 0,\ \text{pic\_width\_in\_luma\_samples} - 1,\ xInt_L + i\ ) \quad (8\text{-}224)$$

$$yA_{i,j} = \text{Clip3}(\ 0,\ \text{pic\_height\_in\_luma\_samples} - 1,\ yInt_L + j\ ) \quad (8\text{-}225)$$

The positions labelled with lower-case letters within un-shaded blocks represent luma samples at quarter-luma-sample fractional locations. The luma location offset in fractional-sample units ($xFrac_L$, $yFrac_L$) specifies which of the generated

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	luma samples at full-sample and fractional-sample locations is assigned to the predicted luma sample value $predSampleLX_L$. This assignment is as specified in Table 8-8. The value of $predSampleLX_L$ is the output. The variables shift1, shift2 and shift3 are derived as follows: – The variable shift1 is set equal to Min(4, $BitDepth_Y$ − 8), the variable shift2 is set equal to 6 and the variable shift3 is set equal to Max(2, 14 − $BitDepth_Y$). Given the luma samples $A_{i,j}$ at full-sample locations ($xA_{i,j}$, $yA_{i,j}$), the luma samples $a_{0,0}$ to $r_{0,0}$ at fractional sample positions are derived as follows: – The samples labelled $a_{0,0}$, $b_{0,0}$, $c_{0,0}$, $d_{0,0}$, $h_{0,0}$ and $n_{0,0}$ are derived by applying an 8-tap filter to the nearest integer position samples as follows: $a_{0,0} = (-A_{-3,0} + 4 * A_{-2,0} - 10 * A_{-1,0} + 58 * A_{0,0} + 17 * A_{1,0} - 5 * A_{2,0} + A_{3,0}) >> shift1$ (8-226) $b_{0,0} = (-A_{-3,0} + 4 * A_{-2,0} - 11 * A_{-1,0} + 40 * A_{0,0} + 40 * A_{1,0} - 11 * A_{2,0} + 4 * A_{3,0} - A_{4,0}) >> shift1$ (8-227) $c_{0,0} = (A_{-2,0} - 5 * A_{-1,0} + 17 * A_{0,0} + 58 * A_{1,0} - 10 * A_{2,0} + 4 * A_{3,0} - A_{4,0}) >> shift1$ (8-228) $d_{0,0} = (-A_{0,-3} + 4 * A_{0,-2} - 10 * A_{0,-1} + 58 * A_{0,0} + 17 * A_{0,1} - 5 * A_{0,2} + A_{0,3}) >> shift1$ (8-229) $h_{0,0} = (-A_{0,-3} + 4 * A_{0,-2} - 11 * A_{0,-1} + 40 * A_{0,0} + 40 * A_{0,1} - 11 * A_{0,2} + 4 * A_{0,3} - A_{0,4}) >> shift1$ (8-230) $n_{0,0} = (A_{0,-2} - 5 * A_{0,-1} + 17 * A_{0,0} + 58 * A_{0,1} - 10 * A_{0,2} + 4 * A_{0,3} - A_{0,4}) >> shift1$ (8-231)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– The samples labelled $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $j_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$ and $r_{0,0}$ are derived by applying an 8-tap filter to the samples $a_{0,i}$, $b_{0,i}$ and $c_{0,i}$ with $i = -3..4$ in the vertical direction as follows: $e_{0,0} = (-a_{0,-3} + 4 * a_{0,-2} - 10 * a_{0,-1} + 58 * a_{0,0} + 17 * a_{0,1} - 5 * a_{0,2} + a_{0,3}) >> \text{shift2}$ (8-232) $i_{0,0} = (-a_{0,-3} + 4 * a_{0,-2} - 11 * a_{0,-1} + 40 * a_{0,0} + 40 * a_{0,1} - 11 * a_{0,2} + 4 * a_{0,3} - a_{0,4}) >> \text{shift2}$ (8-233) $p_{0,0} = (a_{0,-2} - 5 * a_{0,-1} + 17 * a_{0,0} + 58 * a_{0,1} - 10 * a_{0,2} + 4 * a_{0,3} - a_{0,4}) >> \text{shift2}$ (8-234) $f_{0,0} = (-b_{0,-3} + 4 * b_{0,-2} - 10 * b_{0,-1} + 58 * b_{0,0} + 17 * b_{0,1} - 5 * b_{0,2} + b_{0,3}) >> \text{shift2}$ (8-235) $j_{0,0} = (-b_{0,-3} + 4 * b_{0,-2} - 11 * b_{0,-1} + 40 * b_{0,0} + 40 * b_{0,1} - 11 * b_{0,2} + 4 * b_{0,3} - b_{0,4}) >> \text{shift2}$ (8-236) $q_{0,0} = (b_{0,-2} - 5 * b_{0,-1} + 17 * b_{0,0} + 58 * b_{0,1} - 10 * b_{0,2} + 4 * b_{0,3} - b_{0,4}) >> \text{shift2}$ (8-237) $g_{0,0} = (-c_{0,-3} + 4 * c_{0,-2} - 10 * c_{0,-1} + 58 * c_{0,0} + 17 * c_{0,1} - 5 * c_{0,2} + c_{0,3}) >> \text{shift2}$ (8-238) $k_{0,0} = (-c_{0,-3} + 4 * c_{0,-2} - 11 * c_{0,-1} + 40 * c_{0,0} + 40 * c_{0,1} - 11 * c_{0,2} + 4 * c_{0,3} - c_{0,4}) >> \text{shift2}$ (8-239) $r_{0,0} = (c_{0,-2} - 5 * c_{0,-1} + 17 * c_{0,0} + 58 * c_{0,1} - 10 * c_{0,2} + 4 * c_{0,3} - c_{0,4}) >> \text{shift2}$ (8-240) **Table 8-8 – Assignment of the luma prediction sample predSampleLX$_L$** **xFracL**: 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3 **yFracL**: 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3 **predSampleLX$_L$**: A << shift3, d, h, n, a, e, i, p, b, f, j, q, c, g, k, r ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 164-65. **8.5.3.3.3.3 Chroma sample interpolation process** This process is only invoked when ChromaArrayType is not equal to 0. Inputs to this process are: – a chroma location in full-sample units ($xInt_C$, $yInt_C$),

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– a chroma location in eighth fractional-sample units ($xFrac_C$, $yFrac_C$), – the chroma reference sample array $refPicLX_C$. Output of this process is a predicted chroma sample value $predSampleLX_C$

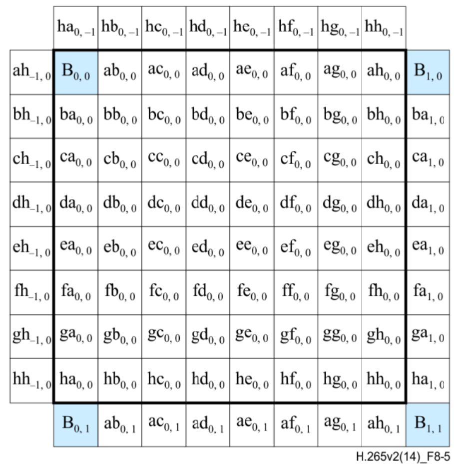


Figure 8-5 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for eighth sample chroma interpolation

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	In Figure 8-5, the positions labelled with upper-case letters $B_{i,j}$ within shaded blocks represent chroma samples at full-sample locations inside the given two-dimensional array $refPicLX_C$ of chroma samples. These samples may be used for generating the predicted chroma sample value $predSampleLX_C$. The locations ($xB_{i,j}$, $yB_{i,j}$) for each of the corresponding chroma samples $B_{i,j}$ inside the given array $refPicLX_C$ of chroma samples are derived as follows: $xB_{i,j} = Clip3(0, (pic\_width\_in\_luma\_samples / SubWidthC) - 1, xInt_C + i)$ (8-241) $yB_{i,j} = Clip3(0, (pic\_height\_in\_luma\_samples / SubHeightC) - 1, yInt_C + j)$ (8-242) The positions labelled with lower-case letters within un-shaded blocks represent chroma samples at eighth-pel sample fractional locations. The chroma location offset in fractional-sample units ($xFrac_C$, $yFrac_C$) specifies which of the generated chroma samples at full-sample and fractional-sample locations is assigned to the predicted chroma sample value $predSampleLX_C$. This assignment is as specified in Table 8-9. The output is the value of $predSampleLX_C$. The variables shift1, shift2 and shift3 are derived as follows: – The variable shift1 is set equal to Min(4, $BitDepth_C$ − 8), the variable shift2 is set equal to 6 and the variable shift3 is set equal to Max(2, 14 − $BitDepth_C$). Given the chroma samples $B_{i,j}$ at full-sample locations ($xB_{i,j}$, $yB_{i,j}$), the chroma samples $ab_{0,0}$ to $hh_{0,0}$ at fractional sample positions are derived as follows: – The samples labelled $ab_{0,0}$, $ac_{0,0}$, $ad_{0,0}$, $ae_{0,0}$, $af_{0,0}$, $ag_{0,0}$ and $ah_{0,0}$ are derived by applying a 4-tap filter to the nearest integer position samples as follows: $ab_{0,0} = (-2 * B_{-1,0} + 58 * B_{0,0} + 10 * B_{1,0} - 2 * B_{2,0}) >> shift1$ (8-243) $ac_{0,0} = (-4 * B_{-1,0} + 54 * B_{0,0} + 16 * B_{1,0} - 2 * B_{2,0}) >> shift1$ (8-244) $ad_{0,0} = (-6 * B_{-1,0} + 46 * B_{0,0} + 28 * B_{1,0} - 4 * B_{2,0}) >> shift1$ (8-245) $ae_{0,0} = (-4 * B_{-1,0} + 36 * B_{0,0} + 36 * B_{1,0} - 4 * B_{2,0}) >> shift1$ (8-246) $af_{0,0} = (-4 * B_{-1,0} + 28 * B_{0,0} + 46 * B_{1,0} - 6 * B_{2,0}) >> shift1$ (8-247)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	$ag_{0,0} = (-2 * B_{-1,0} + 16 * B_{0,0} + 54 * B_{1,0} - 4 * B_{2,0}) >> shift1$ (8-248) $ah_{0,0} = (-2 * B_{-1,0} + 10 * B_{0,0} + 58 * B_{1,0} - 2 * B_{2,0}) >> shift1$ (8-249) – The samples labelled $ba_{0,0}$, $ca_{0,0}$, $da_{0,0}$, $ea_{0,0}$, $fa_{0,0}$, $ga_{0,0}$ and $ha_{0,0}$ are derived by applying a 4-tap filter to the nearest integer position samples as follows: $ba_{0,0} = (-2 * B_{0,-1} + 58 * B_{0,0} + 10 * B_{0,1} - 2 * B_{0,2}) >> shift1$ (8-250) $ca_{0,0} = (-4 * B_{0,-1} + 54 * B_{0,0} + 16 * B_{0,1} - 2 * B_{0,2}) >> shift1$ (8-251) $da_{0,0} = (-6 * B_{0,-1} + 46 * B_{0,0} + 28 * B_{0,1} - 4 * B_{0,2}) >> shift1$ (8-252) $ea_{0,0} = (-4 * B_{0,-1} + 36 * B_{0,0} + 36 * B_{0,1} - 4 * B_{0,2}) >> shift1$ (8-253) $fa_{0,0} = (-4 * B_{0,-1} + 28 * B_{0,0} + 46 * B_{0,1} - 6 * B_{0,2}) >> shift1$ (8-254) $ga_{0,0} = (-2 * B_{0,-1} + 16 * B_{0,0} + 54 * B_{0,1} - 4 * B_{0,2}) >> shift1$ (8-255) $ha_{0,0} = (-2 * B_{0,-1} + 10 * B_{0,0} + 58 * B_{0,1} - 2 * B_{0,2}) >> shift1$ (8-256) – The samples labelled $bX_{0,0}$, $cX_{0,0}$, $dX_{0,0}$, $eX_{0,0}$, $fX_{0,0}$, $gX_{0,0}$ and $hX_{0,0}$ for X being replaced by b, c, d, e, f, g and h, respectively, are derived by applying a 4-tap filter to the intermediate values $aX_{0,i}$ with i = −1..2 in the vertical direction as follows: $bX_{0,0} = (-2 * aX_{0,-1} + 58 * aX_{0,0} + 10 * aX_{0,1} - 2 * aX_{0,2}) >> shift2$ (8-257) $cX_{0,0} = (-4 * aX_{0,-1} + 54 * aX_{0,0} + 16 * aX_{0,1} - 2 * aX_{0,2}) >> shift2$ (8-258) $dX_{0,0} = (-6 * aX_{0,-1} + 46 * aX_{0,0} + 28 * aX_{0,1} - 4 * aX_{0,2}) >> shift2$ (8-259) $eX_{0,0} = (-4 * aX_{0,-1} + 36 * aX_{0,0} + 36 * aX_{0,1} - 4 * aX_{0,2}) >> shift2$ (8-260)

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 11,805,267</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
$$fX_{0,0} = (-4 * aX_{0,-1} + 28 * aX_{0,0} + 46 * aX_{0,1} - 6 * aX_{0,2}) >> \text{shift2} \quad (8\text{-}261)$$
$$gX_{0,0} = (-2 * aX_{0,-1} + 16 * aX_{0,0} + 54 * aX_{0,1} - 4 * aX_{0,2}) >> \text{shift2} \quad (8\text{-}262)$$
$$hX_{0,0} = (-2 * aX_{0,-1} + 10 * aX_{0,0} + 58 * aX_{0,1} - 2 * aX_{0,2}) >> \text{shift2} \quad (8\text{-}263)$$

Table 8-9 – Assignment of the chroma prediction sample predSampleLX_C for (X, Y) being replaced by (1, b), (2, c), (3, d), (4, e), (5, f), (6, g) and (7, h), respectively
<table>
<tr><td>xFracC</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>yFracC</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>predSampleLXC</td><td>B << shift3</td><td>ba</td><td>ca</td><td>da</td><td>ea</td><td>fa</td><td>ga</td><td>ha</td></tr>
<tr><td colspan="9"></td></tr>
<tr><td>xFracC</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>yFracC</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>predSampleLXC</td><td>aY</td><td>bY</td><td>cY</td><td>dY</td><td>eY</td><td>fY</td><td>gY</td><td>hY</td></tr>
</table>

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 165-67.
</td></tr>
<tr><td>[E] obtaining a combined prediction based at least partly upon said first prediction and said second prediction;</td><td>Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding video comprising obtaining a combined prediction based at least partly upon said first prediction and said second prediction.

For example, and without limitation, each of the Accused Products performs a method for decoding video comprising obtaining a combined prediction based at least partly upon said first prediction and said second prediction, corresponding to the decoding process specified by the H.265 Standard. The following specifications provide further evidence of how each of the Accused Products operates:</td></tr>
</table>

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3.4 Weighted sample prediction process **8.5.3.3.4.1 General** Inputs to this process are: – two variables nPbW and nPbH specifying the width and the height of the current prediction block, – two (nPbW)x(nPbH) arrays predSamplesL0 and predSamplesL1, – the prediction list utilization flags, predFlagL0 and predFlagL1, – the reference indices refIdxL0 and refIdxL1, – a variable cIdx specifying colour component index. Output of this process is the (nPbW)x(nPbH) array pbSamples of prediction sample values. The variable bitDepth is derived as follows: – If cIdx is equal to 0, bitDepth is set equal to BitDepth_Y. – Otherwise, bitDepth is set equal to BitDepth_C. The variable weightedPredFlag is derived as follows: – If slice_type is equal to P, weightedPredFlag is set equal to weighted_pred_flag. – Otherwise (slice_type is equal to B), weightedPredFlag is set equal to weighted_bipred_flag. The following applies: – If weightedPredFlag is equal to 0, the array pbSamples of the prediction samples is derived by invoking the default weighted sample prediction process as specified in clause 8.5.3.3.4.2 with the prediction block width nPbW, the prediction block height nPbH, two (nPbW)x(nPbH) arrays predSamplesL0 and predSamplesL1, the prediction list utilization flags predFlagL0 and predFlagL1 and the bit depth bitDepth as inputs. – Otherwise (weightedPredFlag is equal to 1), the array pbSamples of the prediction samples is derived by invoking the weighted sample prediction process as specified in clause 8.5.3.3.4.3 with the prediction block width nPbW, the prediction block height nPbH, two (nPbW)x(nPbH) arrays predSamplesL0 and predSamplesL1, the prediction list utilization flags predFlagL0 and predFlagL1, the reference indices refIdxL0 and refIdxL1, the colour component index cIdx and the bit depth bitDepth as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 167-68.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3.4.2 Default weighted sample prediction process Inputs to this process are: – two variables nPbW and nPbH specifying the width and the height of the current prediction block, – two (nPbW)x(nPbH) arrays predSamplesL0 and predSamplesL1, – the prediction list utilization flags, predFlagL0, and predFlagL1, – a bit depth of samples, bitDepth. Output of this process is the (nPbW)x(nPbH) array pbSamples of prediction sample values. Variables shift1, shift2, offset1 and offset2 are derived as follows: – The variable shift1 is set equal to Max(2, 14 − bitDepth) and the variable shift2 is set equal to Max(3, 15 − bitDepth). – The variable offset1 is set equal to 1 << (shift1 − 1). – The variable offset2 is set equal to 1 << (shift2 − 1). Depending on the values of predFlagL0 and predFlagL1, the prediction samples pbSamples[x][y] with x = 0..nPbW − 1 and y = 0..nPbH − 1 are derived as follows: – If predFlagL0 is equal to 1 and predFlagL1 is equal to 0, the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL0[x][y] + offset1) >> shift1) (8-264) – Otherwise, if predFlagL0 is equal to 0 and predFlagL1 is equal to 1, the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL1[x][y] + offset1) >> shift1) (8-265) – Otherwise (predFlagL0 is equal to 1 and predFlagL1 is equal to 1), the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL0[x][y] + predSamplesL1[x][y] + offset2) >> shift2) (8-266) ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 168.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
[F] decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and	Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding video comprising decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right. For example, and without limitation, of the Accused Products performs a method for decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right. The following specifications provide further evidence of how each of the Accused Products operates: **5.5 Bit-wise operators** The following bit-wise operators are defined as follows: x >> y Arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values of y. Bits shifted into the most significant bits (MSBs) as a result of the right shift have a value equal to the MSB of x prior to the shift operation. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 16.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3.4.2 Default weighted sample prediction process Inputs to this process are: – two variables nPbW and nPbH specifying the width and the height of the current prediction block, – two (nPbW)x(nPbH) arrays predSamplesL0 and predSamplesL1, – the prediction list utilization flags, predFlagL0, and predFlagL1, – a bit depth of samples, bitDepth. Output of this process is the (nPbW)x(nPbH) array pbSamples of prediction sample values. Variables shift1, shift2, offset1 and offset2 are derived as follows: – The variable shift1 is set equal to Max(2, 14 − bitDepth) and the variable shift2 is set equal to Max(3, 15 − bitDepth). – The variable offset1 is set equal to 1 << (shift1 − 1). – The variable offset2 is set equal to 1 << (shift2 − 1). Depending on the values of predFlagL0 and predFlagL1, the prediction samples pbSamples[x][y] with x = 0..nPbW − 1 and y = 0..nPbH − 1 are derived as follows: – If predFlagL0 is equal to 1 and predFlagL1 is equal to 0, the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL0[x][y] + offset1) >> shift1) (8-264) – Otherwise, if predFlagL0 is equal to 0 and predFlagL1 is equal to 1, the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL1[x][y] + offset1) >> shift1) (8-265) – Otherwise (predFlagL0 is equal to 1 and predFlagL1 is equal to 1), the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL0[x][y] + predSamplesL1[x][y] + offset2) >> shift2) (8-266) ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 168.
[G] reconstructing the block of pixels based on the combined precision.	Each of the Accused Products, such as the Hisense 43A7N, performs a method for decoding video comprising reconstructing the block of pixels based on the combined precision.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	For example, of the Accused Products performs a method for decoding video comprising reconstructing the block of pixels based on the combined precision, corresponding to the decoding process specified by the H.265 Standard. The following specifications provide further evidence of how each of the Accused Products operates:

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

	8.5 Decoding process for coding units coded in inter prediction mode **8.5.1 General decoding process for coding units coded in inter prediction mode** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a variable log2CbSize specifying the size of the current coding block. Output of this process is a modified reconstructed picture before deblocking filtering. The derivation process for quantization parameters as specified in clause 8.6.1 is invoked with the luma location (xCb, yCb) as input. The variable $nCbS_L$ is set equal to 1 << log2CbSize. When ChromaArrayType is not equal to 0, the variable $nCbSw_C$ is set equal to (1 << log2CbSize) / SubWidthC and the variable $nCbSh_C$ is set equal to (1 << log2CbSize) / SubHeightC. The decoding process for coding units coded in inter prediction mode consists of the following ordered steps: 1. The inter prediction process as specified in clause 8.5.2 is invoked with the luma location (xCb, yCb) and the luma coding block size log2CbSize as inputs, and the outputs are the array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for the residual signal of coding units coded in inter prediction mode specified in clause 8.5.4 is invoked with the luma location (xCb, yCb) and the luma coding block size log2CbSize as inputs, and the outputs are the array $resSamples_L$ and, when ChromaArrayType is not equal to 0, the arrays $resSamples_{Cb}$ and $resSamples_{Cr}$. 3. The reconstructed samples of the current coding unit are derived as follows: – The picture construction process prior to in-loop filtering for a colour component as specified in clause 8.6.7 is invoked with the luma coding block location (xCb, yCb), the variable nCurrSw set equal to $nCbS_L$, the variable nCurrSh set equal to $nCbS_L$, the variable cIdx set equal to 0, the $(nCbS_L)x(nCbS_L)$ array predSamples set equal to $predSamples_L$ and the $(nCbS_L)x(nCbS_L)$ array resSamples set equal to $resSamples_L$ as inputs. – When ChromaArrayType is not equal to 0, the picture construction process prior to in-loop filtering for a colour component as specified in clause 8.6.7 is invoked with the chroma coding block location (xCb / SubWidthC, yCb / SubHeightC), the variable nCurrSw set equal to $nCbSw_C$, the variable nCurrSh set equal to $nCbSh_C$, the variable cIdx set equal to 1, the $(nCbSw_C)x(nCbSh_C)$ array predSamples set equal to $predSamples_{Cb}$ and the $(nCbSw_C)x(nCbSh_C)$ array resSamples set equal to $resSamples_{Cb}$ as inputs. – When ChromaArrayType is not equal to 0, the picture construction process prior to in-loop filtering for a colour component as specified in clause 8.6.7 is invoked with the chroma coding block location (xCb / SubWidthC, yCb / SubHeightC), the variable nCurrSw set equal to $nCbSw_C$, the variable nCurrSh set

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	equal to $nCbSh_C$, the variable cIdx set equal to 2, the $(nCbSw_C)x(nCbSh_C)$ array predSamples set equal to $predSamples_{Cr}$ and the $(nCbSw_C)x(nCbSh_C)$ array resSamples set equal to $resSamples_{Cr}$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 141-42.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

	8.5.2 Inter prediction process This process is invoked when decoding coding unit whose CuPredMode[xCb][yCb] is not equal to MODE_INTRA. Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a variable log2CbSize specifying the size of the current luma coding block. Outputs of this process are: – an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of luma prediction samples, where $nCbS_L$ is derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below. The variable $nCbS_L$ is set equal to 1 << log2CbSize. When ChromaArrayType is not equal to 0, the variable $nCbSw_C$ is set equal to $nCbS_L$ / SubWidthC and the variable $nCbSh_C$ is set equal to $nCbS_L$ / SubHeightC. The variable $nCbS1_L$ is set equal to $nCbS_L$ >> 1. Depending on the value of PartMode, the following applies: – If PartMode is equal to PART_2Nx2N, the decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. – Otherwise, if PartMode is equal to PART_2NxN, the following ordered steps apply: 1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $nCbS_L$ >> 1 and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, $nCbS_L$ >> 1), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $nCbS_L$ >> 1 and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. – Otherwise, if PartMode is equal to PART_Nx2N, the following ordered steps apply:

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 1, the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to ($nCbS_L$ >> 1, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 1, the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

	– Otherwise, if PartMode is equal to PART_2NxnU, the following ordered steps apply: 1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $nCbS_L >> 2$ and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, $nCbS_L >> 2$), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $(nCbS_L >> 1) + (nCbS_L >> 2)$ and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. – Otherwise, if PartMode is equal to PART_2NxnD, the following ordered steps apply: 1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $(nCbS_L >> 1) + (nCbS_L >> 2)$ and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, $(nCbS_L >> 1) + (nCbS_L >> 2)$), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$, the height of the luma prediction block nPbH set equal to $nCbS_L >> 2$ and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. – Otherwise, if PartMode is equal to PART_nLx2N, the following ordered steps apply: 1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L >> 2$, the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to ($nCbS_L >> 2$, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $(nCbS_L >> 1) + (nCbS_L >> 2)$, the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– Otherwise, if PartMode is equal to PART_nRx2N, the following ordered steps apply: 1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to ($nCbS_L$ >> 1) + ($nCbS_L$ >> 2), the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)$x$(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)$x$(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to ($nCS1_L$ + ($nCbS_L$ >> 2), 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 2, the height of the luma prediction block nPbH set equal to $nCbS_L$ and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)$x$(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)$x$(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. – Otherwise (PartMode is equal to PART_NxN), the following ordered steps apply: 1. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 1, the height of the luma prediction

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	block nPbH set equal to $nCbS_L$ >> 1 and a partition index partIdx set equal to 0 as inputs, and the outputs are an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, two $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 2. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to ($nCbS_L$ >> 1, 0), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 1, the height of the luma prediction block nPbH set equal to $nCbS_L$ >> 1 and a partition index partIdx set equal to 1 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 3. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to (0, $nCbS_L$ >> 1), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 1, the height of the luma prediction block nPbH set equal to $nCbS_L$ >> 1 and a partition index partIdx set equal to 2 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. 4. The decoding process for prediction units in inter prediction mode as specified in clause 8.5.3 is invoked with the luma location (xCb, yCb), the luma location (xBl, yBl) set equal to ($nCbS_L$ >> 1, $nCbS_L$ >> 1), the size of the luma coding block $nCbS_L$, the width of the luma prediction block nPbW set equal to $nCbS_L$ >> 1, the height of the luma prediction block nPbH set equal to $nCbS_L$ >> 1 and a partition index partIdx set equal to 3 as inputs, and the outputs are the modified $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ and, when ChromaArrayType is not equal to 0, the two modified $(nCbSw_C)x(nCbSh_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 142-44

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3 Decoding process for inter prediction samples **8.5.3.3.1 General** Inputs to this process are: – a luma location (xCb, yCb) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture, – a luma location (xBl, yBl) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block, – a variable nCbS specifying the size of the current luma coding block, – two variables nPbW and nPbH specifying the width and the height of the luma prediction block, – the luma motion vectors mvL0 and mvL1,

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	– when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1, – the reference indices refIdxL0 and refIdxL1, – the prediction list utilization flags, predFlagL0, and predFlagL1. Outputs of this process are: – an $(nCbS_L)x(nCbS_L)$ array $predSamples_L$ of luma prediction samples, where $nCbS_L$ is derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below, – when ChromaArrayType is not equal to 0, an $(nCbSw_C)x(nCbSh_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where $nCbSw_C$ and $nCbSh_C$ are derived as specified below. The variable $nCbS_L$ is set equal to nCbS. When ChromaArrayType is not equal to 0, the variable $nCbSw_C$ is set equal to nCbS / SubWidthC and the variable $nCbSh_C$ is set equal to nCbS / SubHeightC. Let $predSamplesL0_L$ and $predSamplesL1_L$ be (nPbW)x(nPbH) arrays of predicted luma sample values and, when ChromaArrayType is not equal to 0, $predSamplesL0_{Cb}$, $predSamplesL1_{Cb}$, $predSamplesL0_{Cr}$ and $predSamplesL1_{Cr}$ be (nPbW / SubWidthC)x(nPbH / SubHeightC) arrays of predicted chroma sample values. For X being each of 0 and 1, when predFlagLX is equal to 1, the following applies: – The reference picture consisting of an ordered two-dimensional array $refPicLX_L$ of luma samples and, when ChromaArrayType is not equal to 0, two ordered two-dimensional arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$ of chroma samples is derived by invoking the process specified in clause 8.5.3.3.2 with refIdxLX as input. – The array $predSamplesLX_L$ and, when ChromaArrayType is not equal to 0, the arrays $predSamplesLX_{Cb}$ and $predSamplesLX_{Cr}$ are derived by invoking the fractional sample interpolation process specified in clause 8.5.3.3.3 with the luma locations (xCb, yCb) and (xBl, yBl), the luma prediction block width nPbW, the luma prediction block height nPbH, the motion vectors mvLX and, when ChromaArrayType is not equal to 0, mvCLX, and the reference arrays $refPicLX_L$, $refPicLX_{Cb}$, and $refPicLX_{Cr}$ as inputs. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 161-62.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.5.3.3.4.2 Default weighted sample prediction process Inputs to this process are: – two variables nPbW and nPbH specifying the width and the height of the current prediction block, – two (nPbW)x(nPbH) arrays predSamplesL0 and predSamplesL1, – the prediction list utilization flags, predFlagL0, and predFlagL1, – a bit depth of samples, bitDepth. Output of this process is the (nPbW)x(nPbH) array pbSamples of prediction sample values. Variables shift1, shift2, offset1 and offset2 are derived as follows: – The variable shift1 is set equal to Max(2, 14 − bitDepth) and the variable shift2 is set equal to Max(3, 15 − bitDepth). – The variable offset1 is set equal to 1 << (shift1 − 1). – The variable offset2 is set equal to 1 << (shift2 − 1). Depending on the values of predFlagL0 and predFlagL1, the prediction samples pbSamples[x][y] with x = 0..nPbW − 1 and y = 0..nPbH − 1 are derived as follows: – If predFlagL0 is equal to 1 and predFlagL1 is equal to 0, the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL0[x][y] + offset1) >> shift1) (8-264) – Otherwise, if predFlagL0 is equal to 0 and predFlagL1 is equal to 1, the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL1[x][y] + offset1) >> shift1) (8-265) – Otherwise (predFlagL0 is equal to 1 and predFlagL1 is equal to 1), the prediction sample values are derived as follows: pbSamples[x][y] = Clip3(0, (1 << bitDepth) − 1, (predSamplesL0[x][y] + predSamplesL1[x][y] + offset2) >> shift2) (8-266) ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 168.

EXHIBIT 10
UNITED STATES PATENT NO. 11,805,267
CLAIM CHART FOR INFRINGEMENT OF CLAIM 19 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 11,805,267	HISENSE ACCUSED PRODUCTS
	8.6.7 Picture construction process prior to in-loop filter process Inputs to this process are: – a location (xCurr, yCurr) specifying the top-left sample of the current block relative to the top-left sample of the current picture component, – the variables nCurrSw and nCurrSh specifying the width and height, respectively, of the current block, – a variable cIdx specifying the colour component of the current block, – an (nCurrSw)x(nCurrSh) array predSamples specifying the predicted samples of the current block, – an (nCurrSw)x(nCurrSh) array resSamples specifying the residual samples of the current block. Depending on the value of the colour component cIdx, the following assignments are made: – If cIdx is equal to 0, recSamples corresponds to the reconstructed picture sample array S_L and the function clipCidx1 corresponds to $Clip1_Y$. – Otherwise, if cIdx is equal to 1, recSamples corresponds to the reconstructed chroma sample array S_{Cb} and the function clipCidx1 corresponds to $Clip1_C$. – Otherwise (cIdx is equal to 2), recSamples corresponds to the reconstructed chroma sample array S_{Cr} and the function clipCidx1 corresponds to $Clip1_C$. The (nCurrSw)x(nCurrSh) block of the reconstructed sample array recSamples at location (xCurr, yCurr) is derived as follows: recSamples[xCurr + i][yCurr + j] = clipCidx1(predSamples[i][j] + resSamples[i][j]) (8-327) with i = 0..nCurrSw − 1, j = 0..nCurrSh − 1 ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 180.